

GEOMETRY ACQUISITION AND GRID GENERATION - RECENT EXPERIENCES
WITH COMPLEX AIRCRAFT CONFIGURATIONS

Timothy D. Gatzke, Walter F. LaBozzetta,
John W. Cooley, and Gregory P. Finfrock
McDonnell Aircraft Company
St. Louis, MO

ABSTRACT

As Computational Fluid Dynamics (CFD) analysis methods continue to mature, the ability to generate a suitable grid for complex configurations has become the pacing item in the application of CFD to engineering problems. The variety of forms and detail present in the geometry definition of real vehicles residing in a computer aided design system compounds the difficulties and contributes to the time required to generate a grid. This paper will discuss important issues involved in working with complex geometry and evaluate approaches that have been taken to address these issues in the McDonnell Aircraft Computational Grid System and related geometry processing tools. The issues that will be addressed include the efficiency of acquiring a suitable geometry definition, the need to manipulate the geometry, and the time and skill level required to generate the grid while preserving geometric fidelity.

INTRODUCTION

As Computational Fluid Dynamics (CFD) analysis methods continue to mature, they are being applied to problems that are more complex, both geometrically and in the physics of the flowfield. For example, the numerical solution of the Navier-Stokes equations is becoming increasingly important for the analysis of advanced inlets and nozzles and their integration with an airframe. The ability to generate a suitable grid has become the pacing item in the application of CFD to these problems. For a complex configuration, constructing the grid, including geometry acquisition and grid generation, can today require from several weeks to several months of effort.

Before beginning grid generation on a complex configuration, accurate configuration geometry must be available in a suitable form. Until recently, most geometry being analyzed was composed of simple analytical shapes, such as circular, ogive, cylindrical, bodies of revolution, that could be easily generated within a computer program. Another source of geometry data was cross section cuts through the vehicle surfaces. These cuts were defined by strings of points. Today, the geometry definition may be stored in any one of several computer aided design (CAD) systems currently in use throughout industry. Acquiring geometry from a CAD system presents an added complication to the CFD analysis process because most CAD geometry databases use surface definitions which are not readily compatible with most grid generation tools. Therefore, it is necessary to manipulate CAD geometry before grid generation can begin. Although this paper will focus on data that is defined in a CAD system, several of the issues discussed apply to any geometry, independent of origin.

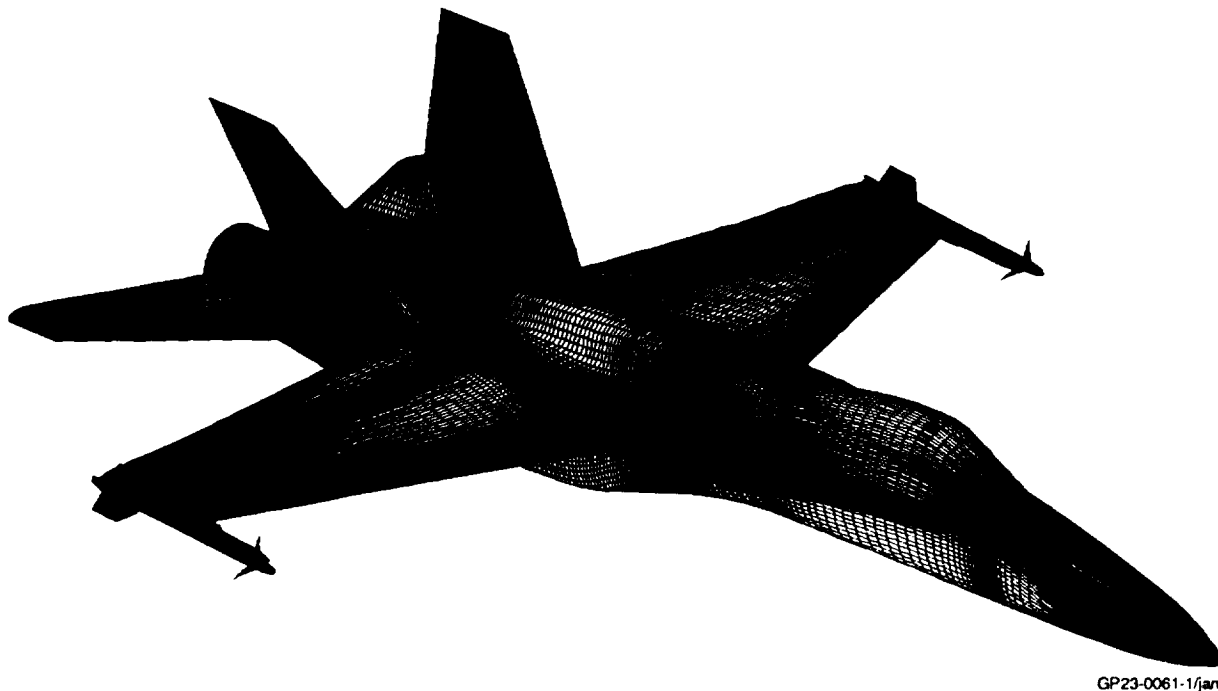
Within the CAD environment, the geometry data can take on varying levels of detail and complexity. The quality of the CAD model can vary as well. Flaws in the model are a practical consideration that must be acknowledged. Also, the CAD system typically provides a large number of different analytic surface types allowing the designer much flexibility in defining vehicle mold lines. This variety of forms and detail, which is present in the geometry definition of real vehicles, compounds the difficulties and contributes to the time required to generate a grid.

There are many issues involved in generating the grid for a complex geometry. These include the efficiency of getting the geometry into a grid generation system, the need to manipulate the geometric data, and the time and skill level required to generate the grid while preserving the fidelity of the geometry. Several approaches have been taken to address these issues in the McDonnell Aircraft Computational Grid System (MACGS) and related geometry processing tools. This paper will discuss the various approaches that have been implemented and identify what has been learned from their implementation.

BACKGROUND

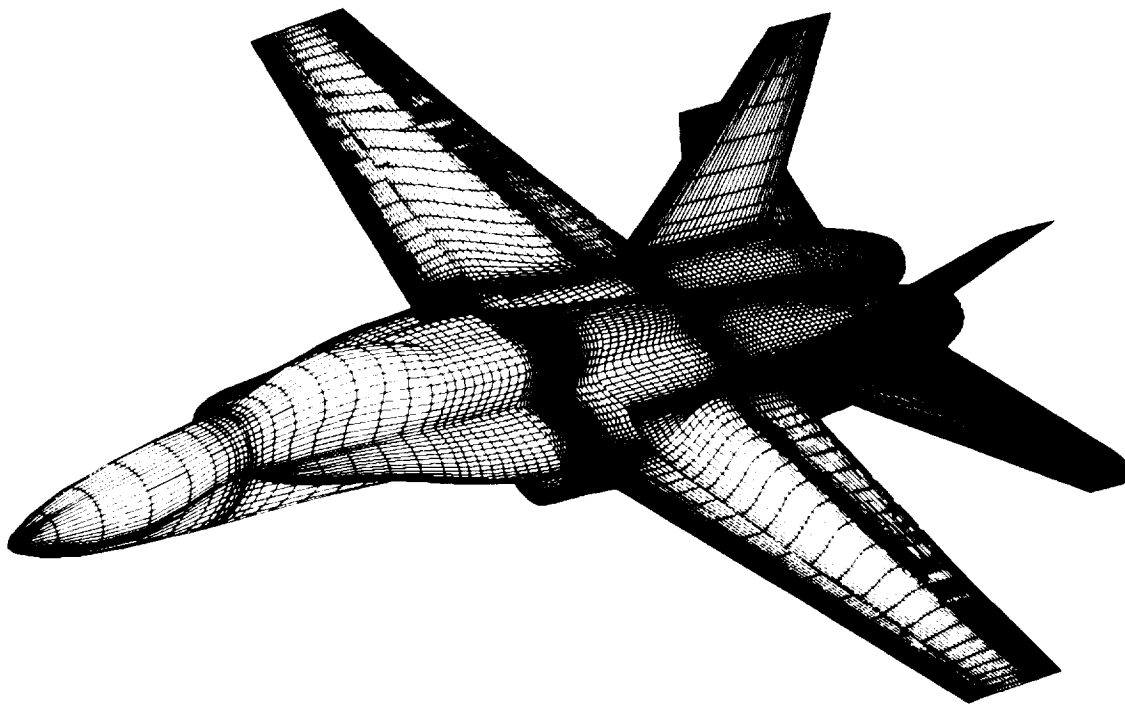
The complexity of geometry has progressed from problems such as wings, isolated forebodies, isolated inlets, and blended body configurations, to full vehicles integrating internal and external flow. Grid generation has progressed from batch methods aimed at a particular configuration, to batch methods able to handle arbitrary configurations, and most recently to interactive graphical tools for complex grid generation tasks. Batch methods for arbitrary topologies, such as EAGLE (Reference 1), gave the user more flexibility than previous methods, but still did not give the user rapid feedback during the decision making process of setting up the batch code inputs. Interactive graphical tools for grid generation, such as the Wright Laboratory (WL) GRIDGEN system (Reference 2), provide improved grids by giving the user more direct control of and feedback from the grid generation process. This is especially important when gridding a new complex configuration. The WL Interactive Graphics for Geometry Generation (I3G) program (Reference 3) was developed to manipulate geometry primarily for input to panel codes and has also been used as the basis for the WL VIRGO grid generation code (Reference 4). These methods have improved grid generation capability but need to address how complex geometry will be obtained.

One of the main modules of MACGS (Reference 5) is based in part on the I3G program. Extensive modifications to I3G at the McDonnell Aircraft Company (MCAIR) have enhanced its use as an arbitrary topology three-dimensional grid generation program, while retaining its ability to manipulate geometry data. MACGS uses an interactive zonal approach which does not require point continuity at zone-to-zone interfaces. This approach is compatible with MCAIR preferred flow solvers. The zonal approach maximizes flexibility and simplifies grid generation by subdividing the physical domain into simpler regions or zones. Figure 1 shows surface grids for a fighter aircraft configuration with a wing tip missile. This Euler grid contained 17 zones with point-continuity at zone interfaces, and required about 3.5 million grid points. In contrast, Figure 2 shows surface grids for a fighter aircraft configuration which included inlet and nozzle geometry. This Navier-Stokes grid contained 17 zones without point-continuity at zone interfaces, and required about 2.6 million grid points. This illustrates that point-mismatch zone interfaces can significantly reduce the number of grid points.



GP23-0061-1/jam

Figure 1. Surface grids for a fighter aircraft configuration with wing tip missile – point continuity at zone interfaces required by the flow solver.



GP23-0061-2/jam

Figure 2. Surface grids for a fighter aircraft configuration with inlet and nozzle – point mismatch at zone interfaces.

In order to facilitate geometry inputs to MACGS, several tools have been developed that allow the user easier access to the CAD geometry database and provide the ability to acquire the geometry data. These include interfaces to specific CAD systems which aid the user in preparing data for the grid generation process, and a tool to process the output of the CAD systems into a form that can be used by the grid generation system.

THE MACGS GRID GENERATION SYSTEM

MACGS is divided into three main modules: a boundary grid generation module, ZONI3G; a field grid generation module, GMAN; and a grid processing module, GPRO. This paper will address approaches that have been implemented in the ZONI3G and GMAN modules. ZONI3G provides the following capabilities:

- 1) Access to geometry data in various formats for interfacing flexibility.
- 2) General surface construction and manipulation tools.
- 3) Surface grid generation tools (algebraic and elliptic) for creating grids on the zone faces.
- 4) A wide range of grid point distribution functions.
- 5) Zone consistency checking for reducing user workload by forcing faces to match at edges (and edges to match at corners) of the zone.

GMAN provides the following capabilities:

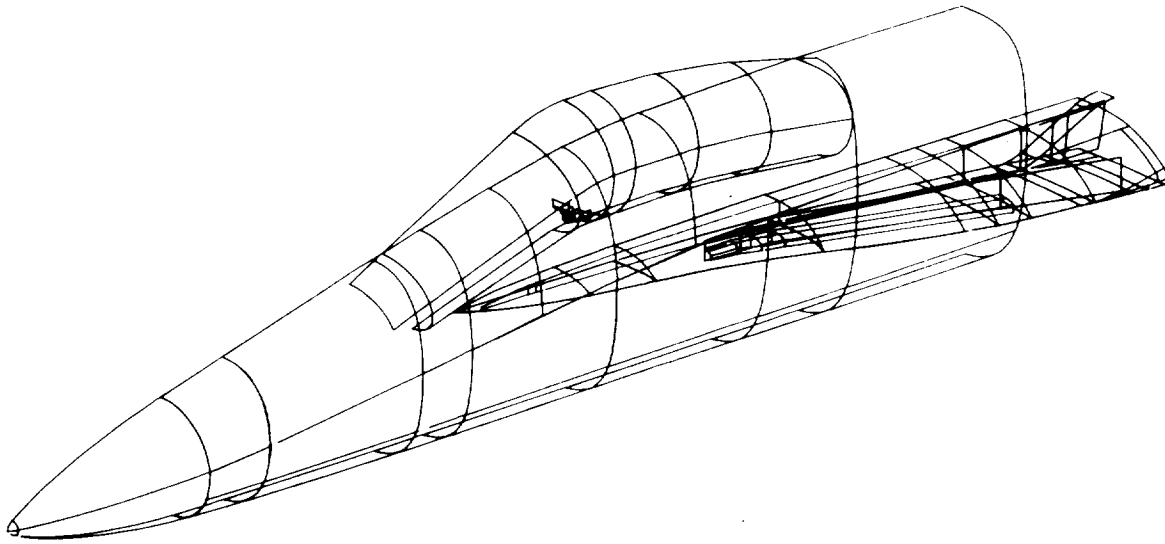
- 1) Algebraic and elliptic three-dimensional field grid generation methods.
- 2) Three-dimensional grid quality assessment tools.
- 3) Interactive specification of boundary conditions on any boundary or subset of a boundary.
- 4) Efficient computation of the interpolation factors for each point on a zone interface of one zone, relative to points located on the zone interface of the adjacent zone (for point-match or point-mismatch interfaces).
- 5) Output formats directly compatible with zonal Euler or Navier-Stokes flowfield prediction codes.

ZONI3G and GMAN are interactive, graphical, menu-driven modules that allow the user to work with geometry and grids in a user-friendly environment. Features of MACGS that affect the processing of geometry and the overall efficiency of the grid generation process will be described in subsequent sections.

GEOMETRY ACQUISITION ISSUES

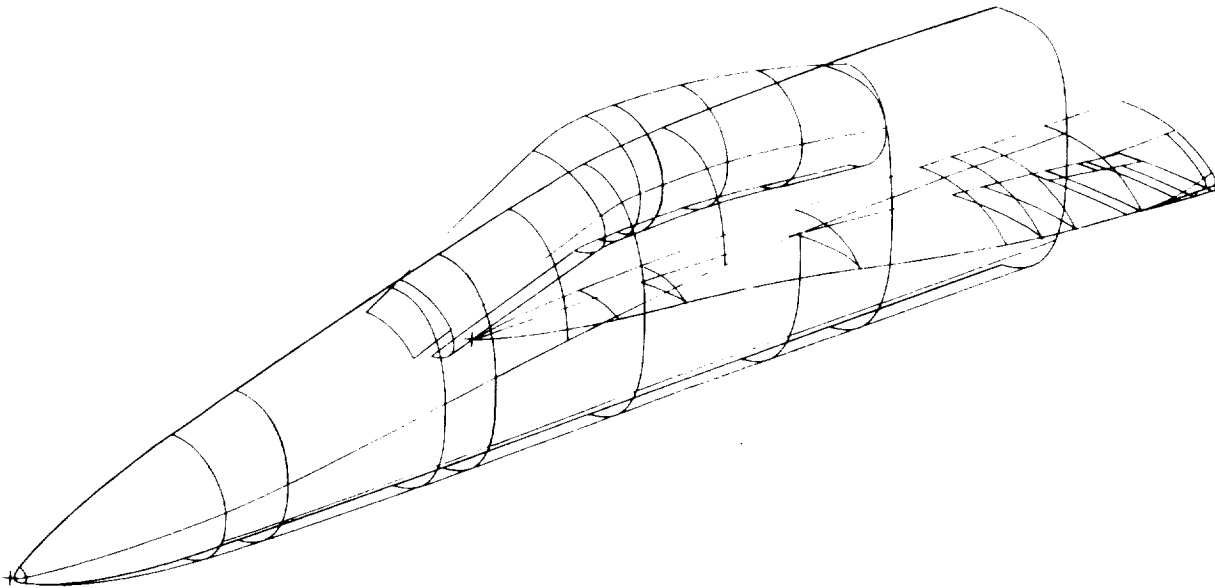
The initial step in the geometry acquisition process is the retrieval and inspection of the CAD model. In many instances, such as the surfaced forebody/LEX model shown in Figure 3, the CAD model contains surface definitions that are unneeded for the CFD analysis. For such a configuration the CAD operator should simplify the model to avoid unwanted section cut definitions in the ensuing step. Simplifying a surface definition may include eliminating surfaces in various areas. This often occurs with protuberances such as antennae, lights, exposed hinges and actuators, or armament fixtures. The CAD operator should have a rough idea of the emphasis of the CFD analysis and should eliminate the protuberances which are nuisances to the engineer generating the grid. If the elimination of protuberances

results in surface holes or mismatch, the CAD operator should correct the definition by creating simple surfaces in these areas. The simplified surfaced model is shown in Figure 4. Modifications to the surfaced model may also be necessary due to incomplete or poor surface definitions. Again the CAD operator should correct the holes and mismatched areas with simple surfaces.



GP23-0061-3/jam

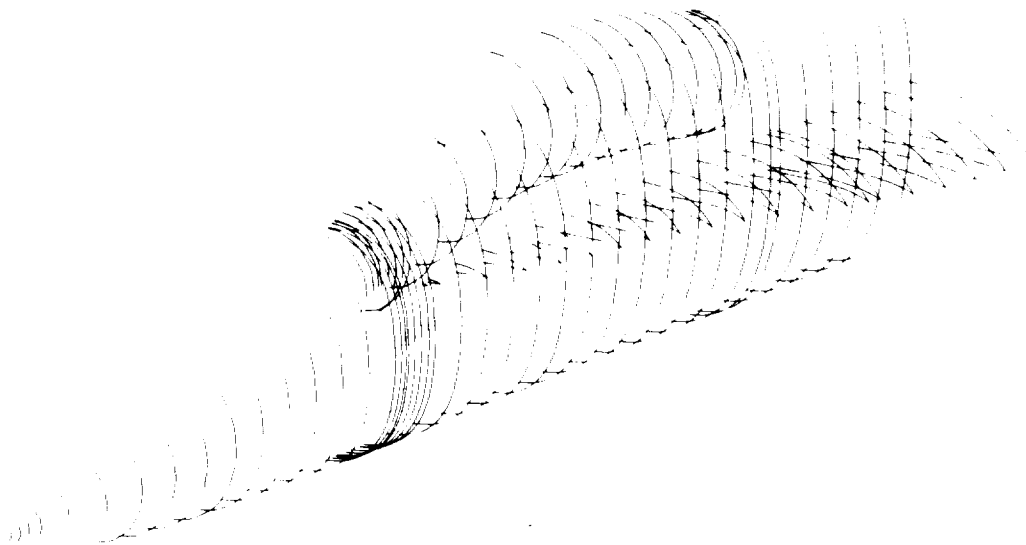
Figure 3. CAD surface model of forebody/LEX configuration.



GP23-0061-4/jam

Figure 4. Simplified CAD definition of forebody/LEX configuration.

For MCAIR CFD applications work, transitioning the CAD model surfaces to point surface definitions is done through the use of an intermediate model definition, composed of sets of analytical curves on the model surfaces. Most of the CAD curve generation functions can be used to generate the sets of curves. Most commonly, these are generated by passing a sequence of cutting planes through a set of surfaces defining the configuration. The user can locate the planes in the orientation that best defines the geometry, and can choose an appropriate spacing interval between planes for the local geometry detail. The number and type of curves created are dependent upon the CAD system, the surface type being cut, and the surface tolerance specified. Often it is necessary to trim curves because they lie on surfaces which intersect (this is usually easier than trimming a CAD surface to an intersection). Figure 5 shows the forebody/LEX configuration with surface curves at several locations. The untrimmed curves illustrate how CAD surfaces intersect one another. Figure 6 shows a set of surface curves, after all curves have been trimmed to the surface intersections.

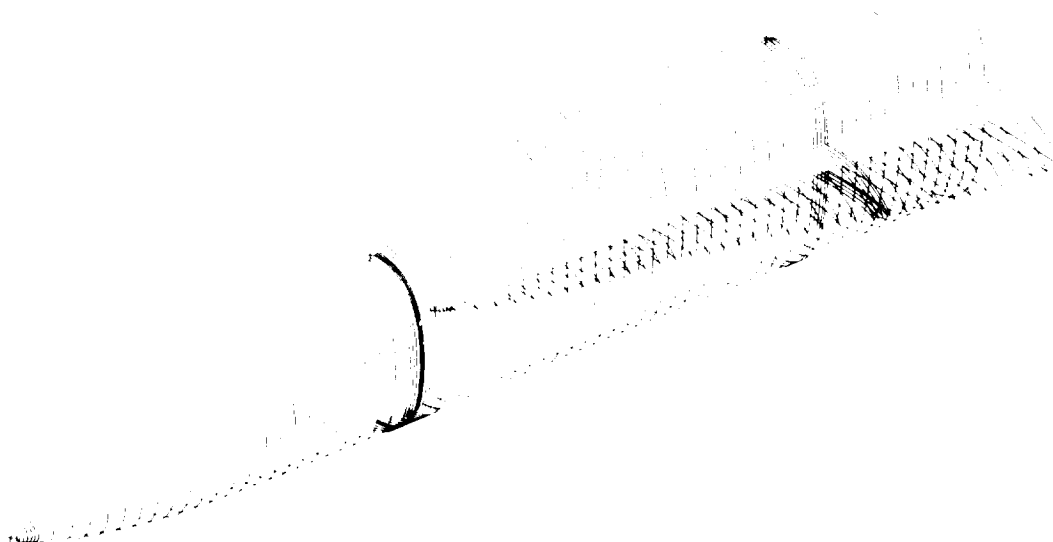


GP23-0061-5/jam

Figure 5. Selected surface curve cuts on CAD definition of forebody/LEX configuration.

Most analytical curves have Initial Graphics Exchange Specification (IGES) (Reference 6) equivalents. Thus, the curve set surface definitions can be transferred from the CAD environment into the grid generation environment through the use of a CAD-to-IGES translator. PCPROC (originally developed at Douglas Aircraft Company and enhanced at MCAIR) is a tool that reads, reorganizes, and outputs curves using the IGES format. IGES entity curve types are converted (at user-controlled tolerance) to parametric cubic spline entities upon being read into PCPROC. Individual curves or entire curve sets can then be merged, sorted, transformed, or otherwise manipulated by using the versatile parametric cubic splines. PCPROC allows the user to generate sets of points on the curve sets using a variety of distribution functions such as equal arc, hyperbolic sine, hyperbolic tangent, and curvature dependent spacing. The end points of curve segments can be preserved, as well as the location of any slope discontinuities. The point surfaces or curves generated can then be written out as a rectangular surface grid for input to MACGS. Once a detailed set of curves has been generated to define a

configuration, it can be saved for future analyses. These sets of curves can then be accessed more quickly than the CAD database if different distributions of points, or some merging of the geometry for a redesigned component such as a canopy, inlet, or wing is required.



GP23-0061-6/jam

Figure 6. Trimmed analytic curve definition of forebody/LEX configuration.

GEOMETRY MANIPULATION ISSUES

Identifying the optimum environment for manipulating geometry into the form desired for the final grid is an important geometry/grid system issue. Depending on the CFD analysis to be performed, the grid for a complex geometry may include faired inlets, nozzles, or boundary layer diverters. It may even involve the removal of a major component such as a wing, with a faired "cap" in its place. Minor modifications such as collapsing finite thickness trailing edges or extending horizontal tail trailing edges to the fuselage intersection must be dealt with on an everyday basis. Experience has shown that one-third to one-half of the time between obtaining point surfaces from CAD and completing the grid is spent manipulating the geometry into the form desired for the application at hand. Figure 7 shows the complex geometry of the nacelle/boundary layer diverter/LEX area of a fighter aircraft. Since the diverter may not be required for a particular analysis, a fairing, as shown in Figure 8, must be defined, maintaining as much of the original geometry definition as possible. Such geometry generation tasks are well suited to being defined in the CAD environment, with its wide range of surface generation capabilities. However, this requires an upstream communication between the CAD operator and the engineer generating the grid to define what faired or filled surfaces must be generated in the CAD system. Often, the CAD geometry is already defined when a new CFD analysis is to begin. The options in this situation are to have a CAD operator fabricate the desired faired surfaces, train the engineer generating the grid to operate the CAD system and let him create the geometry himself, or allow the creation of these surfaces in the "grid generation" phase, using grid generation system tools to modify the geometry acquired from the CAD system.

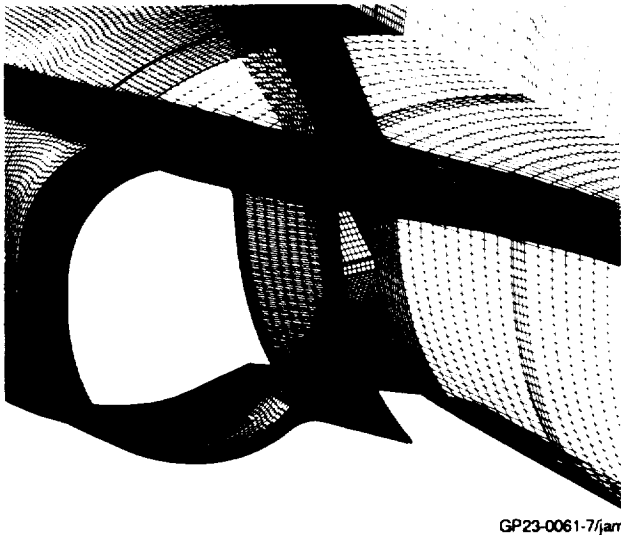


Figure 7. Nacelle/boundary layer diverter/LEX geometry for a fighter aircraft.

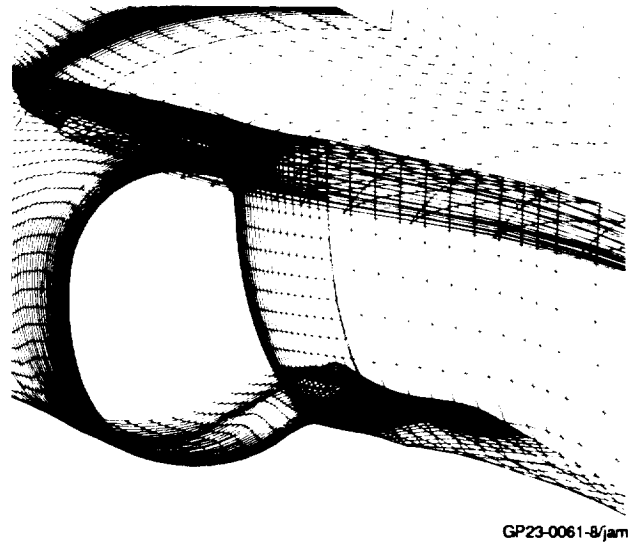


Figure 8. Nacelle/faired boundary layer diverter/LEX grid for a fighter aircraft.

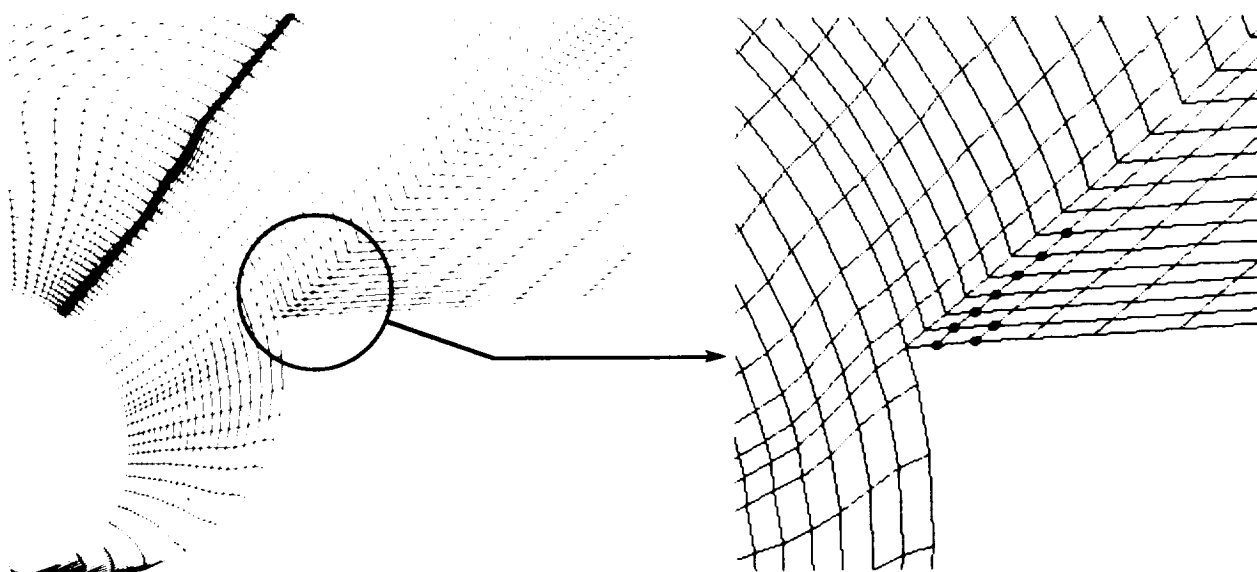
The latter approach is taken in MACGS. CAD geometry of the configuration is modeled during the geometry acquisition phase of the CFD analysis. The resulting analytical curve definition is stored in a central location, to be used as a geometry source for future grids. Depending on the objectives of the current CFD analysis, the engineer may refine the configuration to suit his needs using the point geometry manipulation tools in ZONI3G. If a subsequent CFD analysis on the same vehicle geometry but with different objectives is needed, the original geometry is available without accessing the CAD system.

ZONI3G contains a wide range of geometry manipulation tools (presently for point definition surfaces), allowing surface-to-surface intersections, breaking surfaces at intersection curves, and creating surfaces in space within boundary curves. This approach puts the creation of fairings in the hands of the engineer generating the grid, who best understands what modifications should be made for the analysis at hand. Since the shape and definition of these surfaces may not be critical to the analysis, this approach may be more time and cost effective than creating these surfaces inside the CAD system.

MCAIR experience has shown that CFD engineers prefer having manipulation capability in the grid generation system because the need to modify the geometry may not be apparent until the grid generation process is well underway. However, this approach leaves no clear-cut distinction between the tool which should be used for geometry definition and the one to be used for grid generation. A complex aspect of this approach involves maintaining geometry integrity within an environment in which the user is free to change the shape of components by fairing or filling gaps, but wants to maintain the CAD definition of other components.

Geometry integrity is not as simple as ensuring that surface grid points lie on a CAD database geometry definition. Many of the surfaces to be gridded do not reside in the CAD system in their modified form. Figure 9 shows the area near the junction of a fighter nozzle and horizontal tail. The points highlighted by dots represent grid points lying on an artificial extension of the horizontal tail into the nozzle. These points cannot be referenced to any CAD database surface since in the actual geometry there is a gap between the tail and nozzle in this region.

Giving the CFD engineer the ability to fair and modify geometry within the grid generation tool inherently gives him the ability to lose the same geometry integrity the grid generation system must strive to maintain. As such, this approach assumes that the CFD engineer must take some responsibility for maintaining geometry integrity. The grid generation system alone cannot absolutely ensure that CAD geometry integrity has been maintained. If this is unacceptable, the CFD engineer must have a complete definition of the faired, filled, or otherwise modified geometry to be gridded available in CAD, and must also accept the additional upstream time or training which this requires.



GP23-0061-9/jam

Figure 9. Points on horizontal tail extension not residing in CAD database.

GRID GENERATION ISSUES

Once the geometry has been obtained and is ready for grid generation, the issues to be addressed fall primarily into one of two categories: the preservation of the geometry during the grid generation process, and the overall efficiency of that process. One of the barriers to wider acceptance of CFD in the engineering community is the need to produce usable results in a timely manner (as defined by project goals and schedules). As mentioned previously, interactive methods provide the flexibility and immediate feedback to the user that is necessary for complex geometries. But to meet aggressive schedules, the grid generation system needs to assist the user throughout the process. This includes acquiring the geometry, generating face grids, putting faces together to form zones, generating the field grid, checking the quality of the grid, setting boundary conditions, and calculating interpolation factors at zone interfaces. Automation of much of the process is a key to future gains in grid generation efficiency.

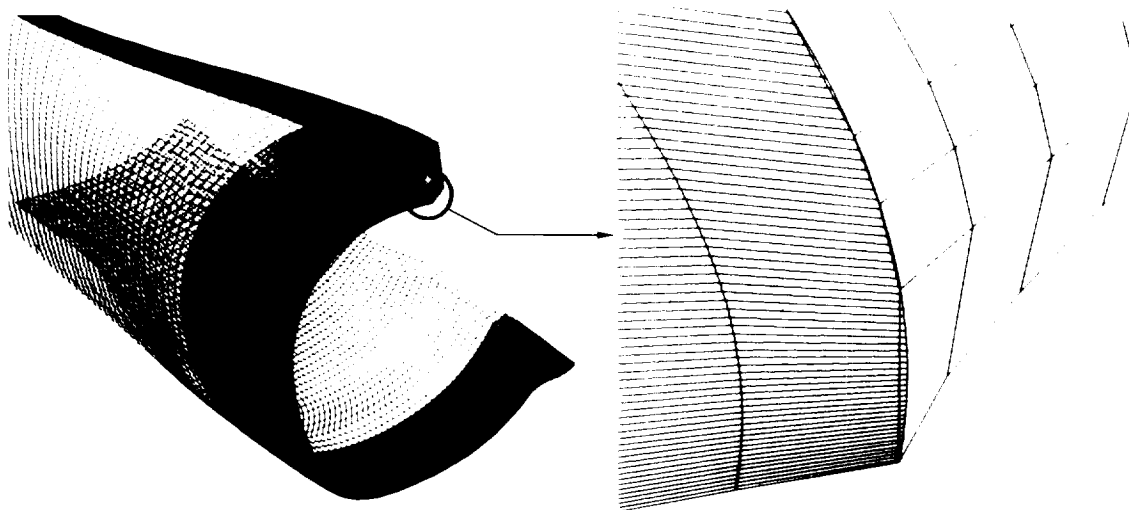
One of the main goals of ZONI3G is to help the user preserve the surface geometry integrity during the grid generation process. One approach is to distinguish between the surface definition and the grid distribution. Once surface geometry has been read into ZONI3G or created within ZONI3G, any subsequent grid

generation operation on that surface will refer back to the original surface definition. This prevents degradation of the surface geometry through successive gridding operations and thus preserves the integrity of the surface geometry. This was implemented by storing two types of surfaces within ZONI3G. The first type is a physical surface which is defined by physical coordinates. This type is used for the initial geometry that is read into the program as well as some surfaces created in the program from scratch. The second type is a parametric surface which is defined by parametric coordinates referencing one or more physical surfaces. All ZONI3G operations can be performed on both types of surfaces.

Operating on a parametric surface has distinct advantages for preservation of the geometry. An operation on the parametric coordinates will return parametric coordinates, ensuring that the new surface is on the original surface definition. This bookkeeping is transparent to the user. No subsequent operations, such as projecting a surface onto a geometry database, are necessary to preserve the geometry.

This approach works well for parametric surfaces that reference a single physical surface. However, after several operations, a parametric surface may reference several physical surfaces. Operations on parametric surfaces referencing multiple physical surfaces are difficult because the parametric coordinates defining the surface reference different coordinate systems, i.e., the local parametric coordinate system of each underlying geometry.

Another difficult area is the location where surfaces with different underlying geometries meet. It must be decided how points that lie between the underlying geometries should be specified and which geometry, if any, they will reference. This problem is apparent in Figure 10 which shows the geometry for a typical nacelle cowl-fuselage intersection. The high curvature of the nacelle cowl leading edge is well represented on the cowl surface, but not on the abutting fuselage surface. Distributing points on the splines of the two surfaces results in a mismatch due to the differing initial curve definition. Resolving these mismatches while maintaining underlying geometry can be very difficult. This issue becomes especially troublesome with viscous grids where a small difference in the geometry can be large compared to the grid point spacing.



GP23-0061-10/jam

Figure 10. Nacelle cowl - fuselage intersection geometry.

Another disadvantage to utilizing parametric surfaces in addition to physical surfaces is that it increases the size of the program, both in lines of source code and memory requirements, and increases development time. This approach can also decrease the efficiency of the system since conversions between physical and parametric surfaces may be required as additional steps in many operations. Further development is needed in order to maintain the geometry throughout the process without sacrificing efficiency.

ZONI3G operates on curves and surfaces. For interpolating between points, a curve is represented by an equation fit through the defining points. For a surface, curves in each direction can be fit. Originally ZONI3G used a nonlinear Akima fit (Reference 7). Since surfaces within ZONI3G are defined as a set of curves, it is not necessary for each curve to have the same number of points. The ability to operate on surfaces made up of curves with different numbers of points on each curve is very useful, since curves defining the geometry that are generated in CAD systems can be represented this way. Grid generation can proceed directly on a surface of this type without requiring an intermediate surface of a different form to be generated. As a result, one level of manipulation that could change the original geometry is eliminated.

For surfaces with the same number of points on each curve, an optional bicubic spline representation was also investigated. This method is the same as that used in EAGLE. One advantage of this representation is that the computations to convert from physical coordinates to parametric coordinates and back are faster. This method is acceptable for surfaces defined by points that are fairly evenly distributed, however the parametric space is highly dependent on the initial point distribution. This dependency can be overcome by scaling the parametric space coordinates using the arc lengths of the curves defining the surface. However, this approach has an associated sacrifice in speed. Even after scaling based on arc lengths, the Akima parametric space is still slightly less dependent on the curves that define the surface. The Akima surface representation also has the advantage of being a local fit. As a result, the surface representation in one area is not affected by the movement of a distant point on the surface. In general, the Akima surface representation produces fewer wiggles than the bicubic spline method and results in a better representation for realistic geometries. For either type of representation, where new distributions of points are specified, it is the local parametric values and the type of parametric space used which are stored, and not the new physical coordinates themselves.

Many complex configurations contain slope discontinuities. A nonlinear fit of these surfaces will model these discontinuities poorly, and introduces wiggles in the surface definition. To address this problem, a linear fit option was added to the surface representations in ZONI3G. A curve can be represented by either a linear or nonlinear fit through the defining points. For a surface, each direction can be represented by either a linear or nonlinear fit through the defining points, as specified by the user. A linear/nonlinear flag is stored in the definition of parametric surface types so that future operations on these surfaces use the appropriate fit. This option improves the integrity of surfaces near a discontinuity and has worked well for geometries containing both smooth surfaces and surfaces with discontinuities.

The efficiency of the process is another key issue in grid generation. The interactive graphical approach is a significant step toward improving efficiency. It is extremely beneficial to see the surface grid point distribution immediately rather than awaiting completion of a batch job. However, the interactive

environment by itself cannot completely solve the efficiency issue. With some previous grid generation codes, the user was responsible for orienting the six computational faces that define a three-dimensional structured grid zone. If the user is not meticulous, several attempts may be needed to get all of the surfaces correct. The ZONI3G zone entity improves three-dimensional zonal grid generation capability by grouping the boundary geometry and grids associated with each of the six computational faces of a zone into a single entity. The process of generating a zone in MACGS includes automated reordering of faces once the zone orientation has been established, and checking for point match along the common edge of adjacent faces.

In some cases, once the user has defined certain faces of a zone, defining the remaining faces is straightforward and the interactive specification of these remaining faces may be tedious. The ZONI3G user can accelerate this process by generating grids for all of the remaining faces of a zone automatically. Missing edges are generated as straight lines between corner points, and then each missing face is interpolated from its edge grids using transfinite interpolation.

For generating the 3-D field grid in the interactive environment, algebraic methods are very attractive because of their speed. On low-end workstations, elliptic methods are often too time-consuming to apply to the entire grid. To reduce time and computer requirements, the GMAN user can apply elliptic methods to a limited local region of the grid where the algebraic methods are inadequate. GMAN uses several algebraic and elliptic methods to locally refine a grid. These operations can be performed over a user-selected range of the grid, with modifications displayed immediately. Algebraic methods are adequate for most of the grid as long as the boundary grids are reasonable. In those cases, local elliptic refinement can be used to efficiently correct most flaws.

Grid quality assessment and correction improves the performance of the entire solution process by eliminating wasted flow solver computation time due to flawed grids and increasing solution convergence rate by providing a higher quality grid. Negative volume, crossed side, collapsed face, and zero volume checks of the QBERT grid evaluation code (Reference 8), developed at WL, have been implemented in GMAN. However, MCAIR experience shows that satisfying these checks is not always sufficient to ensure that the flow solver will run. Additional orthogonality and smoothness measures are being investigated. MCAIR experience has shown that if care is taken to ensure good boundary grids, generating the interior grids of the zones is fairly straightforward and requires minimal grid refinement. Therefore, grid quality checking for surfaces is planned for insertion into ZONI3G.

CONCLUSIONS

Acquiring a suitable geometry definition from a CAD system is a significant part of the overall process of generating a grid. The details that exist in the CAD model and the numerous forms that the surface can take complicate the process. The grid system must work with the CAD surfaces directly, or it must accurately convert analytical surface data in the CAD environment to discrete surface definitions in the grid generation environment.

Geometry manipulation is an integral part of the CFD grid generation process. Geometry manipulation currently occupies up to half of the time required to generate a grid for a complex configuration. Geometry manipulation can be done

either within the CAD system or within the grid generation system. The MACGS approach provides manipulation capability in the grid generation system. Therefore, the CFD engineer need not be an expert in the use of the CAD systems, and can often create the modified surfaces in a more timely manner than waiting for the availability of the CAD operator. Precisely maintaining the geometry integrity of these modifications is not critical. Thus, an associated CAD geometry is not required.

Maintaining geometry integrity of critical surfaces during the grid generation process, especially for the novice at generating grids, must be a concern for any grid generation system. If provisions for automatically maintaining geometry integrity are included in the grid system, as is the case with MACGS, degradation of system efficiency needs to be minimized. After the geometry acquisition, manipulation, and preservation issues have been addressed, automation of the process to generate the grid is the critical factor. Automation of some grid generation steps has been shown to reduce grid generation time by 20-25%. Interactive operation will continue to be important to give the user direct control throughout the grid generation process. Improvement and/or automation in areas such as grid quality assessment techniques is needed to provide further reductions in grid generation time.

ACKNOWLEDGEMENTS

This research was partially supported by the McDonnell Douglas Independent Research and Development Program.

REFERENCES

1. Thompson, J. F.: Program EAGLE User's Manual. AFATL-TR-88-117, Vols. I-III, Sept. 1988.
2. Steinbrenner, J. P.; Chawner, J. R.; and Fouts, C. L.: The GRIDGEN 3D Multiple Block Grid Generation System. WRDC-TR-90-3022, Vol. I, II, July 1990.
3. LaBozzetta, W. F.; Cole, P. E.; Kreis, R. I.; and Finfrock, G. P.: Configuration Data Management System - System Documentation. AFWAL-TR-87-3064, Vol. I, II, December 1987.
4. Amdahl, Lt. D. A.: Interactive Multi-Block Grid Generation. The Second International Conference on Numerical Grid Generation in Computational Fluid Dynamics. Miami Beach, Fl., November 1988.
5. Gatzke, T. G.; LaBozzetta, W. F.; Finfrock, G. P.; Johnson, J. A.; and Romer, W. W.: MACGS: A Zonal Grid Generation System for Complex Aero-Propulsion Configurations. AIAA-91-2156, June 1991.
6. Smith, B.; and Wellington, Jr.: Initial Graphics Exchange Specification (IGES), Version 3.0. NBSIR-86-3359, April 1986.
7. Akima, H.: A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. Journal of the Association for Computing Machinery, vol. 17, no. 4, October 1970, pp. 589-602.
8. Strang, W. Z.: QBERT: A Grid Evaluation Code. AFWAL-TM-88-193, July 1988.

